This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

Efficient Continuous Scanning in RFID Systems

Bo Sheng
Northeastern University
Email: shengbo@ccs.neu.edu

Qun Li
College of William and Mary
Email: liqun@cs.wm.edu

Weizhen Mao
College of William and Mary
Email: wm@cs.wm.edu

***Abstract*—RFID is an emerging technology with many potential applications such as inventory management for supply chain. In practice, these applications often need a series of continuous scanning operations to accomplish a task. For example, if one wants to scan all the products with RFID tags in a large warehouse, given a limited reading range of an RFID reader, multiple scanning operations have to be launched at different locations to cover the whole warehouse. Usually, this series of scanning operations are not completely independent as some RFID tags can be read by multiple processes. Simply scanning all the tags in the reading range during each process is inefficient because it collects a lot of redundant data and consumes a long time. In this paper, we develop efficient schemes for continuous scanning operations defined in both spatial and temporal domains. Our basic idea is to fully utilize the information gathered in the previous scanning operations to reduce the scanning time of the succeeding ones. We illustrate in the evaluation that our algorithms dramatically reduce the total scanning time when compared with other solutions.**

I. Introduction

Radio frequency identification (RFID) is a promising and highly desirable technology for many applications such as object tracking, electronic passport, inventory control and supply chain management. In these applications, every item is attached with an RFID tag, which contains an ID describing the item. The ID information on each RFID tag can be collected by an RFID reader through wireless channels. Compared with conventional bar-codes, RFID technology offers several attractive features, such as remote and multiple access, simple computational ability, and large rewritable on-tag memory. While it is believed that people will greatly benefit from RFID technology, some important problems remain unsolved and seriously hinder the practical implementation of RFID systems.

The fundamental operation in RFID systems is to scan and read the ID information from every tag. In this operation, time efficiency is of crucial importance for many RFID applications, especially when they deal with a large volume of RFID tags. For example, inventory management often requires an RFID reader to scan all the products in a warehouse. This task usually involves tens of thousands of RFID tags, assuming every product is attached with one tag, and mandates a quick scanning operation to be feasibly implemented.

Previous literature has mainly focused on developing efficient protocols for a single scanning operation. In practice, however, many tasks cannot be accomplished by a single scan. Instead, multiple scanning operations have to be continuously launched. For example, to scan all the products in a large warehouse, it is impossible for an RFID reader to read all the tags at one location due to the limited reading range. Usually a single mobile reader (or multiple readers) has to launch multiple scanning operations at different locations to cover all the tags in the entire warehouse.

Continuous scanning is generally defined as a series of multiple scanning operations. In this paper, we study continuous scanning in both spatial and temporal domains. The above warehouse example represents a *spatial continuous scanning*, where a series of scans are executed at different locations. Namely, *temporal continuous scanning* represents a series of scans occurring at different time points. It is often used for monitoring inventory update. For example, some applications may want to keep track of the products stored at a certain location in a dynamic environment where new products may be put on shelf and some existing products may be moved out. An RFID reader, in this case, has to periodically scan all the present RFID tags to keep a fresh record.

In both spatial and temporal continuous scanning, an important property is that each individual scan usually is not completely independent. Some of the scanning operations may involve overlapping RFID tags. In spatial continuous scanning, the processes at adjacent locations inevitably have some tags in common in the reading ranges. Similarly, in temporal continuous scanning, some products may stay in the reading range for a long time, thus can be read by several consecutive scans. Therefore, the simple solution that scans all the RFID tags in the reading range is inefficient because each scan may collect a lot of redundant information which has already been gathered in the previous scans.

This paper proposes efficient algorithms for continuous scanning without collecting all the IDs. Our main idea is to take advantage of the information previously gathered about the inventory and only collect the IDs of the newly added tags and remove the IDs that are no longer present. We make the following contributions. (1) We design two efficient algorithms for continuous scanning, one to collect the 'new' tag IDs and the other to detect those RFID tags that have been moved out. (2) For each algorithm, we first make the assumption that the difference of two overlapped sets is bounded. Later, we relax this assumption and discuss the algorithms in a more general setting. (3) We conduct extensive simulations and illustrate the time efficiency of the proposed algorithms.

The rest of this paper is organized as follows. We review the related work in Section II and introduce some preliminary information about RFID systems in Section III. We formulate the problem in Section IV. We propose our main algorithms in Section V and discuss some extensions in Section VI.

978-1-4244-5837-0/10/$26.00 ©2010 IEEE

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

Finally, we evaluate the proposed algorithms in Section VII and conclude in Section VIII.

II. Related Work

RFID technology has been considered in many applications [1]–[5]. Previous research has focused on anti-collision protocols, which can be classified into two major categories. The first category is ALOHA-like protocols [6]–[17]. We will review the details in Section III-A. The second category is based on tree traversal technique [18]–[23], where a binary tree is built with all possible tag data values as the leaves and each non-leaf node represents the prefix bit string of its descendants. The RFID reader then traverses this tree by broadcasting the prefix and finally obtains all the data values.

In our problem of continuous scanning, however, collision is not the key challenge. In the evaluation, we will show that typical anti-collision protocols are not suitable for this problem. Our solutions are still based on the ALOHA protocol, but specifically designed for continuous scanning. Adaptive splitting proposed in [24] is the only existing approach that can be applied to our problem. We will compare it with our solution in the evaluation and show that tree traversal is time consuming in a large scale system with long IDs (e.g., 96bit ID). Furthermore, some recent work has extracted useful information based on the typical ALOHA scheme, e.g., counting the number of RFID tags [25]–[27]. However, this paper considers a more general task of collecting tag IDs. The intuition behind our scheme of checking the existing tags in Section V-B is similar to the Bloom filter, which has been well studied in the literature [28]–[33]. However, our algorithm has a different goal of minimizing the scanning time with a certain accuracy requirement. The protocol and the corresponding analysis are also quite different.

III. RFID Background and Our System Model

This section presents the background information about RFID and some basic components in our system model.

A. Slotted ALOHA Protocol

An RFID system consists of RFID readers and tags. Each RFID tag can store data and communicate with readers. The most common passive RFID tags have no battery supply, and transmit data by reflecting the received signals from readers.

Using wireless channels, RFID systems suffer from data collision. When more than one tags respond to a reader at the same time, none of their data can be successfully received. Slotted ALOHA is a popular anti-collision protocol implemented by major RFID manufacturers. We briefly review the protocol in this subsection because our design is built upon it. In this protocol, the RFID reader first broadcasts a number f to all tags indicating the following time *frame* is divided to f *slots*. After receiving f, each tag will randomly pick a slot index from 0 to $f-1$ and load the index into a slot counter (sc). Usually, RFID tags use a pseudo-random number generator, which takes a random seed from the reader and hashes the random seed with tag IDs to a slot index. Let r be the random seed sent by the reader, x be the tag ID, the slot index and the initial value of sc will be $hash(x||r)$ mod f. In the rest of this paper, we use $h_f(x,r)$ to represent this operation.

The reader then sequentially scans every slot in the frame. It uses a 'slot end' command to close the current slot and start the next slot, which also triggers every tag to decrease its slot counter (sc) by one. If a tag's sc becomes zero, it will backscatter its ID in the coming slot. From the RFID reader's view, there are three possible scenarios for each slot. First, if only one tag T replies in a slot (the slot is called *single-reply slot*), the reader will send an acknowledgment of success (ACKS) to notify T that the data is successfully received. Tag T will then keep silent (inactive) in the rest of the session. Second, if multiple tags respond in the same slot, the reader will detect a signal collision (the slot is called *collision slot*). The reader will then send an acknowledgment of failure (ACKF) to notify the responding tags (>1) that it has failed to receive the data. These tags will keep active. Third, if no tag responds in a slot (called *empty slot*), the reader will close the slot immediately. At the end of a frame, if collisions have occurred in this frame, the reader will start a new frame, in which only the active tags will participate. Fig. 1 illustrates the state diagram of an RFID tag in the slotted ALOHA protocol.

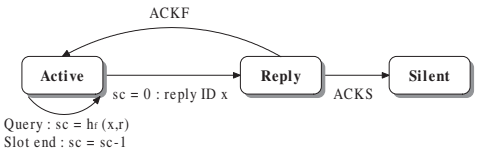


Fig. 1: State Diagram of an RFID tag with ID x: There are three states and each directional edge is annotated by the transition condition or a pair of 'condition: action'. 'sc' denotes the slot counter and 'Query' includes the frame size f and random seed r.

B. Modified ALOHA Protocol

In this paper, we make a minor modification to the slotted ALOHA protocol, so that we can use it not only to collect IDs from RFID tags, but also to efficiently select a set of tags as details will be explained later. The only difference from the typical protocol is the transition from state 'Active' to state 'Reply'. In our design, when $sc=0$, the tag has three options for the action, which is controlled by the reader. In the 'slot end' command, the reader adds two bits as an opcode, which instructs each tag with $sc=0$ to act accordingly as follows. (1) **no reply:** If the opcode is 0, the tag will not reply. (2) **short reply:** If the opcode is 1, the tag will reply with a random short binary string. The length can be as short as possible provided the reader can detect the collision while more than one tag respond. Usually, the short reply is less than 10 bits. (3) **reply ID:** If the opcode is 2, the tag will reply with its ID as the typical protocol.

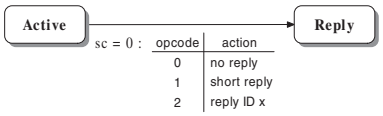


Fig. 2: Modified Transition from 'Active' to 'Reply': The transition action when $sc=0$ depends on the opcode in the previous 'slot end' command.

The modified state diagram of a tag is illustrated in Fig. 2. Note extra hardware design is needed to support this modified protocol. However, based on the functionality of current RFID tags, this modification can be easily implemented.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

C. Slot Timing

In the slotted ALOHA protocol, timing of each slot, which is the time duration of the slot, is an important parameter for calculating total scanning time. In our algorithms, there are different slot timings in accordance with three possible actions of RFID tags as mentioned earlier. First, for the action of 'no reply', we use t_{nr} to denote the slot timing. Second, for the action of 'short reply', we may have three different scenarios, empty slot, single-reply slot and collision slot. The timings of single-reply and collision slot are the same, indicated as t_{sr}, and the timing of an empty slot is shorter, indicated as t_{em}. Third, 'reply ID' action results in the same three scenarios as 'short reply'. The timing for an empty slot is t_{em} and the timing for single-reply and collision slot is t_{ID}. Table I summarizes the notations of slot timing used in this paper. These timings are hardware-dependent parameters, which will be quantified in the evaluation. A common relation is $t_{nr} < t_{em} < t_{sr} \ll t_{ID}$.

	Empty	Single-Reply	Collision
No Reply	t_{nr}		
Short Reply	t_{em}	t_{sr}	t_{sr}
Reply ID	t_{em}	t_{ID}	t_{ID}

TABLE I: Slot Timings

D. Optimal Frame Size for Collecting IDs

Our scheme also uses the typical slotted ALOHA protocol to collect IDs from all active RFID tags. Thus, we need set an appropriate frame size in order to achieve time efficiency.

Assume there are n active tags and f is the frame size in the typical ALOHA. We will successfully collect an ID only in a single-reply slot. For a certain slot, the probability of being an a single-reply slot is $p_1 = \frac{n}{f} \cdot (1-\frac{1}{f})^{n-1} = \frac{n}{f} \cdot e^{-\frac{n}{f}}$. Thus, there is a single-reply slot every $\frac{1}{p_1}$ slots on average. Considering that empty and non-empty slots have different time durations, these $\frac{1}{p_1}$ slots take a time of $\frac{1}{p_1}(p_0 \cdot t_{em} + (1-p_0) \cdot t_{ID})$, where p_0 is the probability for a slot to be an empty slot, $p_0 = (1-\frac{1}{f})^n = e^{-\frac{n}{f}}$. The optimal f should minimize the above formula, which gives the minimum scanning time per ID collected. The above formula can be regarded as a function of $\frac{n}{f}$ denoted as $g(\frac{n}{f})$. Once t_{em} and t_{ID} are set, we can derive the optimal f by solving the derivative equation $g'(\frac{n}{f}) = 0$.

As mentioned in the related work, a lot of previous work has derived the 'optimal' frame size. However, they all regard the number of slots as the measurement of scanning time, ignoring the varying timings for different types of slots. Our analysis here is more practical and accurate.

IV. Problem Formulation

In this section, we formulate our problem more precisely. In spatial continuous scanning, our objective is to efficiently collect the newly introduced RFID tags at each location. In temporal continuous scanning, besides collecting 'new' tags, our objective also includes efficiently detecting the RFID tags that have been moved away.

Essentially, continuous scanning boils down to a general task with two overlapping sets of RFID tags, $\mathcal{S}$ and $\overline{\mathcal{S}}$, at each scanning location/time-point. Set $\mathcal{S}$ contains all RFID tags in the current reading range. The other set $\overline{\mathcal{S}}$ is the previously gathered RFID tags that may be present in the current reading range, usually the RFID tags collected in the adjacent locations/time-points. Our problem is to scan set $\mathcal{S}$ with the knowledge of $\overline{\mathcal{S}}$. We first define two important types of tags:

- **Unknown tags:** the present RFID tags whose IDs have not been collected in the previous scanning operations, i.e., $\mathcal{S}-\overline{\mathcal{S}}$. Let $U = |\mathcal{S} - \overline{\mathcal{S}}|$.
- **Missing tags:** the tags that have been previously scanned, but no longer exist, i.e., $\overline{\mathcal{S}} - \mathcal{S}$. Let $M = |\overline{\mathcal{S}} - \mathcal{S}|$.

Thus, our goal is to efficiently collect IDs from *unknown tags* and for some applications (temporal continuous scans), remove *missing tags* from the record.

Our solutions are based on randomized algorithms, thus cannot guarantee perfect accuracy. Some unknown tags may not be collected by our schemes and some missing tags may not be detected either. Additionally, for any accuracy requirement, a deterministic guarantee cannot be provided. Therefore, we describe the accuracy constraint as follows. Let U' and M' respectively be the numbers of uncollected unknown tags and undetected missing tags *after* applying our schemes ($U' \leq U$ and $M' \leq M$). Smaller U' and M' represent more accurate results. We quantify the accuracy by making them upper bounded by two parameters R_U and R_M respectively with a probability higher than α.

To summarize, given the previously obtained inventory $\overline{\mathcal{S}}$, two upper bounds R_U and R_M, and a probability $\alpha \in (0, 1]$, we would like to use the minimal time to scan $\mathcal{S}$, such that

$$\mathbf{Pr}(U' \leq R_U) > \alpha \text{ and } \mathbf{Pr}(M' \leq R_M) > \alpha.$$

In addition, our schemes are based on an assumption that the difference between $\overline{\mathcal{S}}$ and $\mathcal{S}$, characterized by U and M, is bounded. We use U_{max} and M_{max} to denote the bounds, i.e., $U = |\mathcal{S} - \overline{\mathcal{S}}| \leq U_{max}$ and $M = |\overline{\mathcal{S}} - \mathcal{S}| \leq M_{max}$. Note that U_{max} and M_{max} could be loose estimations. We will show the impacts of their accuracy in evaluation. In Section VI, we relax this assumption and discuss a general setting without prior knowledge about the difference between the two overlapping sets. Note that our problem setting is for a particular single scanning operation in a series of continuous scanning and all these parameters may vary at different locations or time points.

V. Solutions

To achieve our objective, we need accomplish the following two tasks with a probability of at least α:

1) To collect $\geq U - R_U$ unknown tags from $\mathcal{S} - \overline{\mathcal{S}}$;
2) To detect $\geq M - R_M$ missing tags from $\overline{\mathcal{S}} - \mathcal{S}$.

The rest of the section describes two algorithms, one for each task in the problem.

A. Collect Unknown Tags

The hurdle to efficiently collecting the unknown tags is that the tags in $\overline{\mathcal{S}} \cap \mathcal{S}$, which have already been read previously, will interfere and make the responses from the unknown tags buried

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

in overwhelming "noises". Once we can suppress the known tags from responding while keeping the unknown tags active, applying the typical slotted ALOHA protocol would efficiently collect the unknown tags with no interference. Therefore, we propose a two-phase algorithm consisting of the **selecting phase** and **collecting phase**. In the selecting phase, we select *only* unknown tags and keep them active. Note that the selected unknown tags could be a subset of $\mathcal{S} - \overline{\mathcal{S}}$. In other words, we suppress and inactivate all the known tags in $\overline{\mathcal{S}} \cap \mathcal{S}$ as well as partial unknown tags. In the collecting phase, we simply use the typical slotted ALOHA to scan all active tags.

The selecting phase is the key step in the algorithm and our design is based on the existing slotted ALOHA. As we have introduced, in the slotted ALOHA, each tag randomly picks a slot in a frame based on the hashed value of its ID with a random seed sent by the reader. Since we know the set $\overline{\mathcal{S}}$, the random seed r, and the hash function, it is easy to determine whether a particular slot would be occupied by a known tag. When checking the slots one by one, we signal the tags replying in those slots to be silent and keep the tags replying in other slots active. However, some slots may be occupied by both known and unknown tags and our method will mistakenly miss those unknown tags. To counteract this, therefore, we run the two-phase algorithm using different random seeds for several rounds so that all unknown tags may be exposed.

Algorithm 1 $CU(f, \delta)$: **C**ollect **U**nknown Tags

1: $\mathcal{I} \leftarrow \overline{\mathcal{S}}$, $p \leftarrow 1$
2: **while** $p \geq \delta$ **do**
3: Reader powers on and generates a random seed r
4: Reader calculates $PE(\mathcal{I}, f, r)$
5: $p \leftarrow p \cdot (1 - \frac{PE(\mathcal{I},f,r)}{f})$
6: Reader broadcasts f and r to tags
7: Each tag with ID x responds in slot $h_f(x, r)$
8: **for** slot $i = 0$ to $f - 1$ **do**
9: Reader sets $opcode = 0$ at the end of slot $i - 1$
10: **if** slot i is a pre-empty slot **then**
11: Reader sends ACKF
12: **else**
13: Reader sends ACKS
14: **end if**
15: **end for**
16: Reader collects active tags by the typical slotted ALOHA ($opcode = 2$) and adds their IDs to $\mathcal{I}$
17: Reader powers off
18: **end while**
19: Return $\mathcal{I}$

Algorithm 1 (called the CU scheme) illustrates the protocol taking two parameters f and δ. How to determine the parameters is discussed later in this subsection. $\mathcal{I}$ is the resulting inventory which is expected to include sufficient unknown tags. The protocol first sets $\mathcal{I} = \overline{\mathcal{S}}$ and iteratively update $\mathcal{I}$ in the successive rounds. Inside the while loop, lines 3-15 present the selecting phase and line 16 is for the collecting phase with details omitted. In the selecting phase, the reader first generates a random seed r and broadcasts f and r. Each tag takes f as the frame size and picks the slot (from 0 to $f - 1$) to respond computed by the hashed value of its ID and r (line 7). A slot is called a ***pre-empty slot*** if no known tag responds in that slot. We use $PE(\mathcal{I}, f, r)$ to denote the number of pre-empty slots after hashing all tag IDs in $\mathcal{I}$ with r onto f slots, that is,

$$PE(\mathcal{I}, f, r) = |\{j | j \in [0, f), \forall x \in \mathcal{I}, h_f(x, r) \neq j\}|.$$

Being aware of the known tags, the reader can emulate the hashing for each known tag ID in $\overline{\mathcal{S}}$ and determine if a particular slot is pre-empty. The reader then sends commands to make the tags replying in the pre-empty slots stay active in the following stage while other tags are kept silent. In the collecting phase, the reader simply collects the IDs of all active tags via the typical slotted ALOHA protocol and adds them to $\mathcal{I}$. In Algorithm 1, since an unknown tag randomly picks a slot to respond in each round, it has $\frac{PE(\mathcal{I},f,r)}{f}$ probability to be collected. We use a variable p to keep track of the probability that an unknown tag will not be identified after the current round (line 5). Obviously, p becomes smaller as the reader runs more rounds of queries. The process terminates when $p < \delta$.

The performance of our protocol heavily depends on the parameters f and δ. The following theorems show how to properly set these two parameters. Our objective is to minimize the scanning time while the accuracy requirement is satisfied. Note that our objective here only considers the scanning time spent in the selecting phase because the collecting phase is affected not by the settings of f and δ, but by the number of newly collected RFID tags related to the requirement parameter R_U. In the next, Theorem 1 gives the condition that guarantees the accuracy requirement. Theorem 2 provides a bound for the expected number of the iterations in Algorithm 1 and Theorem 3 bounds the scanning time in the selecting phase by a function of the frame size. The results are utilized to find the optimal parameter f in Theorem 4.

Theorem 1: After running Algorithm 1, we can guarantee $\mathbf{Pr}(U' \leq R_U) > \alpha$ if we set

$$\delta \leq \frac{R_U + a}{2a + U_{max}} - \frac{\sqrt{a \cdot (a \cdot U_{max} + 2U_{max} \cdot R_U - 2R_U^2)}}{\sqrt{U_{max}} \cdot (2a + U_{max})},$$

where $a = (\Phi^{-1}(\alpha))^2$ and Φ is the standard normal CDF.

Proof: Recall that U and U' represent the number of unknown tags before and after running Algorithm 1. For each unknown tag before Algorithm 1, the event that it will not be collected by Algorithm 1 is independent and the probability of the event is $p < \delta$ according to the algorithm. Therefore, U' follows a binomial distribution $U' \sim B(U, p)$. Consider another binomially distributed random variable $z \sim B(U_{max}, \delta)$. Since $p < \delta$ and $U \leq U_{max}$, $\mathbf{Pr}(U' < R_U) > \mathbf{Pr}(z < R_U)$. When U_{max} is large, z is approximated by a normal distribution[1], $z \sim N(\mu, \sigma^2)$, where $\mu = \delta \cdot U_{max}$ and $\sigma^2 = \delta \cdot (1 - \delta) \cdot U_{max}$. Therefore, based on the property of normal distribution, $\mathbf{Pr}(z < R_U) = \Phi(\frac{R_U - \mu}{\sqrt{2}\sigma})$. After plugging δ in the hypothesis, we have $\mathbf{Pr}(z < R_U) \geq \alpha$. Thus, $\mathbf{Pr}(U' < R_U) > \alpha$. ■

Theorem 2: Algorithm 1 is expected to terminate after $k = \ln \delta / \ln(1 - e^{-\frac{|\overline{\mathcal{S}}| + U_{max}}{f}})$ rounds.

[1]This approximation helps quickly calculate the CDF.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

Proof: In each round (loop variable in line 2) in Algorithm 1, each slot (loop variable in line 8) in the frame is a pre-empty slot if no tag in $\mathcal{I}$ selects the slot, which has a probability of $(1-\frac{1}{f})^{|\mathcal{I}|}$. Thus, the expected value of $PE(\mathcal{I},f,r)$ is $f\cdot(1-\frac{1}{f})^{|\mathcal{I}|}$. Since $|\mathcal{I}|\leq|\overline{\mathcal{S}}|+U_{max}$ in any round, the expected value of $\frac{PE(\mathcal{I},f,r)}{f}$ is

$$\frac{f\cdot(1-\frac{1}{f})^{|\mathcal{I}|}}{f}\geq(1-\frac{1}{f})^{|\overline{\mathcal{S}}|+U_{max}}=e^{-\frac{|\overline{\mathcal{S}}|+U_{max}}{f}}.$$

After k rounds, the value of p in Algorithm 1 will be

$$\prod_{1}^{k}(1-\frac{PE(\mathcal{I},f,r)}{f})<(1-e^{-\frac{|\overline{\mathcal{S}}|+U_{max}}{f}})^{k}<\delta.$$

Thus, the algorithm will terminate after k rounds. ■

Theorem 3: The expected scanning time of the selecting phase in Algorithm 1 is bounded by $ST_c=k\cdot f\cdot t_{nr}$, where k is the number of rounds in Algorithm 1.

Proof: Algorithm 1 spends totally $k\cdot f$ slots in the selecting phase. In each slot, the reader sets the opcode to 0 in the 'slot end' command. Thus, the timing of each slot in the selecting phase is t_{nr}. ■

Theorem 4: The optimal value of the frame size is $f=1.443\cdot(|\overline{\mathcal{S}}|+U_{max})$.

Proof: With Theorem 2 and 3, the expected scanning time of the selecting phase is bounded by $ST_c=\ln\delta/\ln(1-e^{-\frac{|\overline{\mathcal{S}}|+U_{max}}{f}})\cdot f\cdot t_{nr}$. By solving $\frac{\partial ST_c}{\partial f}=0$, we obtain the optimal value of $f=1.443\cdot(|\overline{\mathcal{S}}|+U_{max})$. ■

B. Detect Missing Tags

As we know, temporal continuous scanning may need remove the missing tags from the inventory. In this subsection, we propose a scheme to detect missing tags and analyze the parameter setting.

Our solution is still based on the slotted ALOHA and utilizes the deterministic randomness in the slot selection of each RFID tag. The basic intuition is to observe the empty slots during the slotted ALOHA and check with the known tag set $\overline{\mathcal{S}}$. Provided that a slot which should be occupied by a known tag becomes empty, we know the tag corresponding to that slot is missing. However, some missing tags may be mapped to a slot with other present tags, and thus will not be detected in the protocol yielding inaccurate results. In our design, we repeat the protocol for multiple rounds (with all tags shuffled in each round) to achieve the accuracy requirement.

Our solution is illustrated in Algorithm 2 (called the DM scheme). $\mathcal{I}$ represents the returned inventory which is supposed to have removed the missing tags after Algorithm 2. $\mathcal{I}'$ represents the set of known tags we have not checked yet. Initially, both $\mathcal{I}$ and $\mathcal{I}'$ are set to $\overline{\mathcal{S}}$. In each round inside the while loop, the reader first generates a random seed r and broadcasts f and r to tags. Each tag randomly picks a slot to respond according to the hash value of its ID. A slot is called a ***pre-single slot*** if only one tag in $\mathcal{I}'$ responds in the slot. With the knowledge of the known tags, the reader can calculate the indexes of the pre-single slots. The reader then checks each pre-single slot to determine if the known tag mapping to the slot is still present (x in line 9). If the pre-single slot is empty, the corresponding known tag ID x will be removed from $\mathcal{I}$. If there is a reply, the reader will make no change on the inventory $\mathcal{I}$ and send ACKS to keep the responding tags silent in the following steps. Let $PS(\mathcal{I}',f,r)$ be the number of pre-single slots yielded by $\mathcal{I}',f$, and r. Thus, in each round, we check the existence of $PS(\mathcal{I}',f,r)$ known tags. The whole algorithm terminates when $|\mathcal{I}'|=0$, i.e., we have checked all the known tags.

Algorithm 2 $DM(f)$: **D**etect **M**issing Tags

```
1: I ← S̄, I' ← S̄
2: Reader powers on
3: while |I'| > 0 do
4:     Reader generates a random number r
5:     Reader broadcasts f and r to tags
6:     Each tag with ID x responds at slot h_f(x, r)
7:     for slot i = 0 to f − 1 do
8:         if slot i is a pre-single slot then
9:             Reader sets opcode = 1 at the end of slot i − 1
10:            Find x ∈ I such that h_f(x, r) = i
11:            I' ← I' − {x}
12:            if no tags respond then
13:                I ← I − {x}
14:            else
15:                Reader sends ACKS
16:            end if
17:        else
18:            Reader sets opcode = 0 at the end of slot i − 1
19:            Reader sends ACKF
20:        end if
21:    end for
22: end while
23: Return I
```

In Algorithm 2, only one parameter f, the frame size, needs to be determined. To set the optimal value to f, we analyze the performance in the following theorems. Our objective is still to minimize the scanning time while the accuracy requirement is satisfied. Theorem 5 presents the condition that guarantees the accuracy requirement for the number of the undetected missing tags (M'). The scanning time in this algorithm, however, cannot be expressed by f in a closed form. Therefore, we present in Lemma 1 a program to estimate the number of iterations executed by Algorithm 2, and then this number is used to express the scanning time in Theorem 6. With Theorem 5 and Theorem 6, we can find the optimal value for f by enumerating all possible values.

Theorem 5: After running Algorithm 2, we can guarantee $\mathbf{Pr}(M'\leq R_M)>\alpha$ if $f>-\frac{U_{max}}{\ln(1-\theta)}$, where

$$\theta=\frac{R_M+a}{2a+M_{max}}-\frac{\sqrt{a\cdot(a\cdot M_{max}+2M_{max}\cdot R_M-2R_M^2)}}{\sqrt{M_{max}}\cdot(2a+M_{max})},$$

$a=(\Phi^{-1}(\alpha))^2$, and Φ is the standard normal CDF.

Proof: In Algorithm 2, a missing tag is not detected only if some unknown tags respond in the pre-single slot it belongs to. Let q be the probability for this scenario, then

$$q=1-(1-\frac{1}{f})^{|\mathcal{S}-\mathcal{I}|}<1-e^{-\frac{U_{max}}{f}}<\theta.$$

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

Recall M and M' are the number of missing tags before and after running Algorithm 2 respectively. Since detecting each missing tag is independent, M' follows a binomial distribution $M' \sim B(M, q)$. Consider another binomially distributed random variable $z \sim B(M_{max}, \theta)$. Since $q < \theta$ and $M \leq M_{max}$, $\mathbf{Pr}(M' < R_M) > \mathbf{Pr}(z < R_M)$. When M_{max} is large, z is approximated by a normal distribution, $z \sim N(\mu, \sigma^2)$, where $\mu = \theta \cdot M$ and $\sigma^2 = \theta \cdot (1-\theta) \cdot M_{max}$. Therefore, based on the property of normal distribution,

$$\mathbf{Pr}(z < R_M) = \Phi(\frac{R_M - \theta \cdot M_{max}}{\sqrt{2\theta \cdot (1-\theta) \cdot M_{max}}}) = \alpha.$$

Thus, $\mathbf{Pr}(M' < R_M) > \alpha$. ■

Lemma 1: Given f, we can estimate the number of iterations executed in Algorithm 2.

Proof: Let n_i be the number of unchecked known tags after round i. In round i, for each slot, the probability that only one known tag in $\mathcal{I}$ selects the slot is $\frac{n_{i-1}}{f}(1-\frac{1}{f})^{n_{i-1}-1} = \frac{n_{i-1}}{f} \cdot e^{-\frac{n_{i-1}}{f}}$. Thus, the expected value of n_i is

$$\begin{aligned} n_{i-1} - PS(\mathcal{I}', f, r) &= n_{i-1} - f \cdot \frac{n_{i-1}}{f} \cdot e^{-\frac{n_{i-1}}{f}} \\ &= n_{i-1} \cdot (1 - e^{-\frac{n_{i-1}}{f}}) \end{aligned}$$

We iteratively calculate n_i until $n_i < 1$, and k is estimated as $\min\{i | n_i < 1\}$. ■

Theorem 6: Assuming Algorithm 2 terminates after k rounds, the scanning time is bounded by

$$ST_d = |\overline{\mathcal{S}}| \cdot t_{sr} + (k \cdot f - |\overline{\mathcal{S}}|) \cdot t_{nr}.$$

Proof: Given k and f, the expected total number of slots is $k \cdot f$. Since every known tag is checked in a pre-single slot, there are $|\overline{\mathcal{S}}|$ pre-single slots and the remaining $k \cdot f - |\overline{\mathcal{S}}|$ slots are no reply slots (lines 18-19 in Algorithm 2) each with duration t_{nr}. For each pre-single slot, there might be a response (with duration t_{sr}) or no response (with duration $t_{em} < t_{sr}$). Therefore, the expected total scanning time is bounded by ST_d as defined. ■

Finally, based on the analysis above, we can enumerate all possible values of f (f is a bounded value in practice), and find the optimal value with the minimum scanning time ST_d.

VI. Extension

Our previous analysis is based on two parameters U_{max} and M_{max}. In some applications, however, it may not be easy to estimate these two bounds or the estimations could be too loose to be meaningful. In this section, we present extensions to our schemes without the assumption of knowing U_{max} and M_{max}. Due to the page limit, we only present the extension for CU and the extension for DM follows a similar fashion.

Our CU scheme is an iterative process and we collect some unknown tags in each round. In this extension, we utilize the information of the collected unknown tags to estimate U_{max}. Thus, our basic idea is to first set a rough estimation for U_{max}, and then iteratively revise it as our scheme proceeds. We expect the estimation of U_{max} converges towards the tighter bound as more iterations are executed.

The details of this extension is presented in Algorithm 3. Initially, we set U_{max} to $|\mathcal{S}|$, the total number of present tags, assuming all the tags are unknown tags. This number $|\mathcal{S}|$ can be obtained by the previous work [25] with a small overhead. We define another variable t to count how many unknown tags have been collected since the algorithm starts.

In line 3, we set the frame size f as we discussed in Theorem 4 and keep updating p as in Algorithm 1. Recall that p is the probability for an unknown tag not to be collected after the current round. The selecting phase and collecting phase are the same as in Algorithm 1. We use variable c to denote the number of newly collected unknown tags and correspondingly update t in line 6. Lines 7-12 are the key part for this extension, which derives a new value for U_{max} based on the observations.

Algorithm 3 Collect Unknown Tags (Extension)

1: $U_{max} \leftarrow |\mathcal{S}|, t \leftarrow 0, p \leftarrow 1$
2: **while** $U_{max} - t > R_U$ **do**
3: $f \leftarrow 1.443 \cdot (|\overline{\mathcal{S}}| + U_{max})$
4: Use f in the selecting phase and execute lines 6-16 in Alg. 1
5: Assume we collect c unknown tags in this round
6: $p \leftarrow p \cdot (1 - \frac{PE(\mathcal{I}, f, r)}{f})$, $t \leftarrow t + c$, $tolp \leftarrow 0$
7: $sum \leftarrow \sum_{x=1}^{|\mathcal{S}|} \binom{x}{t} p^{x-t}(1-p)^t$, $x \leftarrow 1$
8: **while** $tolp < \alpha$ **do**
9: $tolp \leftarrow tolp + \binom{x}{t} p^{x-t}(1-p)^t / sum$
10: $x \leftarrow x + 1$
11: **end while**
12: $U_{max} \leftarrow x$
13: **end while**

The following Theorem 7 and Theorem 8 guarantee the accuracy requirement after running Algorithm 3.

Theorem 7: At any iteration, $\mathbf{Pr}(U \leq U_{max}) > \alpha$.

Proof: At the beginning of the algorithm, there are U unknown tags. In each iteration, variable p records the probability that a certain unknown tag will have not been collected after the current iteration. Thus, the probability of collecting t unknown tags is $\binom{U}{t} p^{U-t}(1-p)^t$. In line 7, variable sum is the summary of the probability of collecting t unknown tags considering all possible values of U. We use sum as a normalizer in the following analysis. Based on the observation of t, $\mathbf{Pr}(U = x) = \binom{x}{t} p^{x-t}(1-p)^t / sum$. In lines 8-12, we set the new value for U_{max} and guarantee that it satisfies the condition $\sum_{x<U_{max}} \mathbf{Pr}(U = x) > \alpha$, which is equivalent to $\mathbf{Pr}(U < U_{max}) > \alpha$. ■

Theorem 8: After running Algorithm 3, we can guarantee $\mathbf{Pr}(U' \leq R_U) > \alpha$.

Proof: Recall that U' is defined as the number of remaining unknown tags, thus $U' = U - t$. According to Theorem 7, $\mathbf{Pr}(U \leq U_{max}) > \alpha$, then we have $\mathbf{Pr}(U' \leq U_{max} - t) > \alpha$. Since Algorithm 3 terminates when $U_{max} - t \leq R_U$, $\mathbf{Pr}(U' \leq R_U) \geq \mathbf{Pr}(U' \leq U_{max} - t) > \alpha$. ■

VII. Evaluation

Our evaluation is based on simulation. Here are some general parameter settings.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

Slot Timings: We keep the following ratios based on hardware and protocol specifications from major RFID manufactures:

$$t_{nr} : t_{em} : t_{sr} : t_{ID} = 1 : 1.5 : 3 : 30.$$

We set the unit time above to 0.25ms [34]. This value may vary for different hardware, but usually appears in the same scale.

Optimal Frame Size for Collecting IDs: According to the slot timing above and the analysis in Section III-D, the optimal frame size for collecting n active tags is $f = 3.48n$. In the collecting phase of our CU scheme, n is estimated as the expected number of the selected unknown tags, $n = \frac{PE}{f} \cdot U_{max}$.

Accuracy Confidence: α is set to 0.99 in all our simulation.

Other Solutions for Comparison: We consider the following three solutions for comparison:

- *Collect All (CA):* The reader collects all tags via the typical slotted ALOHA protocol, ignoring the knowledge of $\overline{\mathcal{I}}$. The initial frame size is set to $|\mathcal{S}|$.
- *Suppress Known (SK):* The reader first broadcasts the known tag IDs one by one. The tag whose ID matches the broadcast ID will be suppressed to keep silent. The reader then uses the ALOHA protocol to collect the remaining active tags.
- *Adaptive Query Splitting (AQS):* This is a tree-traversal-based scheme. We refer the interested reader to [24] for details.

In the rest of this section, we present the scanning time of our schemes and compare with other solutions. Spatial and temporal continuous scanning are separately evaluated. We conduct 100 independent trials for each parameter setting in the simulation, and illustrate the average values. The scanning time deviations of our schemes (< 0.2 second) and other solutions (< 0.5 second) are very small and omitted in the figures. In addition, in all our tested cases, the resulting inventories obtained by our algorithms always satisfy the accuracy requirement.

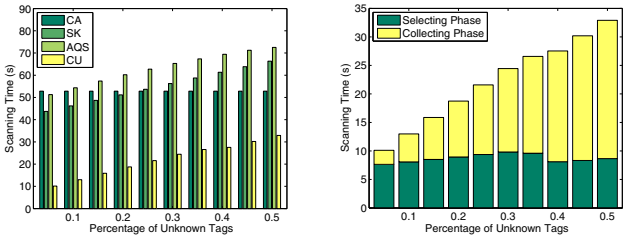
A. Spatial Continuous Scanning

First, we consider the scenario with two overlapping sets of tags. Then, we simulate a more complete case involving a series of continuous scanning operations. Finally, we examine the CU extension scheme proposed in Section VI.

1) Two overlapping sets: The default parameter setting is as follows. There are 5000 tags in both sets, i.e., $|\overline{\mathcal{S}}| = |\mathcal{S}| = 5000$. The number of unknown tags U is set to $U = 0.1 \cdot |\mathcal{S}|$, which means 10% of current tags are unknown tags. For a certain value of U, the accuracy requirement R_U is set to $R_U = 0.1 \cdot U$, i.e., keeping 10% unknown tags uncollected is tolerated. In addition, U_{max} (the upper bound of U) is set to $1.2 \cdot U$, i.e., 20% more than the actual value of U.

Varying U: We first examine the performance with different number of unknown tags, which indicates the difference between the current set of RFID tags and that in the previous scanning. Fig. 3(a) compares our CU scheme with other three solutions, where the horizontal axis represents the ratio of $\frac{U}{|\mathcal{S}|}$.

In Fig. 3(a), the CA scheme keeps the same performance and the other three schemes all yield a linearly increasing performance when U increases because they all contain a collect-unknown-tags process whose performance is proportional to U. The SK scheme becomes worse than the CA scheme when $\frac{U}{|\mathcal{S}|} \geq 0.25$. For a large U, the benefit from collecting unknown tags without active known tags cannot compensate the overhead of suppressing the known tags. The AQS scheme is worse than the SK scheme because in an RFID system with a large ID domain (96bit long ID), tree traversal scheme is very costly.

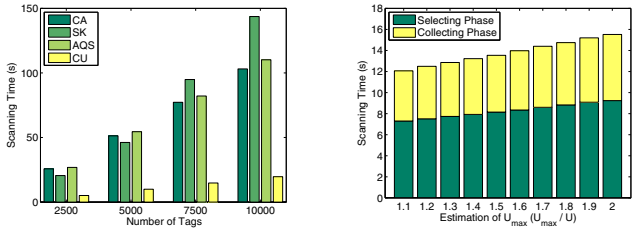


(a) Overall Scanning Time (b) Detailed Scanning Time

Fig. 3: Varying U: (a) The overall scanning time (b) The scanning time spent on the selecting phase and the collecting phase. In our settings, $|\mathcal{S}| = 5000, U/|\mathcal{S}| = \{5\%, 10\%, \ldots, 50\%\}$.

The performance of our CU scheme is superior to all other solutions. For example, in the default setting with $\frac{U}{|\mathcal{S}|} = 0.1$, CU needs less than 13 seconds to finish while SK and CA need 46 and 53 seconds respectively. In the worst tested case with $\frac{U}{|\mathcal{S}|} = 0.5$, CU still saves more than 40% scanning time compared to the CA scheme.

Fig. 3(b) further illustrates the scanning time of the CU scheme spent in the selecting phase and the collecting phase. The collecting phase consumes more time when U increases because it need collect more unknown tags. On the other hand, the scanning time of the selecting phase does not change too much ranging from 7 to 9 seconds. It is interesting to observe that the scanning time of the selecting phase is not monotonously increasing with U.



(a) Varying $|\mathcal{S}|(|\overline{\mathcal{S}}|)$ (b) Varying estimations of U_{max}

Fig. 4: The scanning time of spatial continuous scanning with different varying parameters. (a) The performance of the CA scheme is also presented as a comparison. $|\overline{\mathcal{S}}| = |\mathcal{S}| = \{2500, 5000, 7500, 10000\}$. (b) $U = 500$ and the ratio of U_{max}/U is set to $\{1.1, 1.2, \ldots, 2\}$.

Varying $|\overline{\mathcal{S}}|$ and $|\mathcal{S}|$: In our simulation, we always set $|\overline{\mathcal{S}}| = |\mathcal{S}|$. Both $|\overline{\mathcal{S}}|$ and $|\mathcal{S}|$ indicate the scale of the RFID tags in the simulated scenarios. In this part, we present and compare the performance with varying number of RFID tags. We test four cases, where the number of RFID tags ranges from 2500 to 10000 with 2500 as the interval. Other parameters are derived by keeping the same ratio as in the default setting.

We compare the scanning time in Fig. 4(a). The performance of all schemes is proportional to the number of RFID tags. In all four cases, the scanning time of the CU scheme is less than 24% of that consumed by the CA scheme. The simulation results indicate that the CU scheme can significantly improve

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

the performance in terms of scanning time, especially when dealing with a large amount of RFID tags. For example, when $|\mathcal{S}| = 10000$, the CA scheme needs 104s to finish while the CU scheme consumes about 24s.

Varying U_{max}: Our problem formulation assumes that U_{max} is the estimated bound for U. Here we examine the impact of U_{max} on the performance. In the default setting, the actual number of unknown tags is 500. We vary U_{max} in our tests from 550 (10% more than the actual number) to 1000 (100% more than the actual number). The results are illustrated in Fig. 4(b), where the horizontal axis is the value of $\frac{U_{max}}{U}$ ranging from 1.1 to 2. We observe that a more accurate estimation slightly improves the performance, essentially in the selecting phase. The difference between the best case and worst case in Fig. 4(b) is less than 4 seconds. Therefore, our CU scheme can perform well even with a rough estimation of U_{max}.

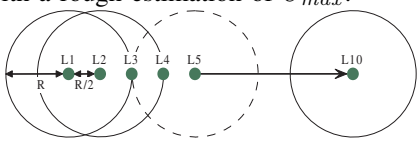


Fig. 5: R is the reading range of the RFID reader, which moves along a straight line and stop at 10 locations, $\{L1, L2, \ldots, L10\}$, to launch the scans. The interval distance between any two consecutive locations is $R/2$.

2) A series of scanning operations: In the next, we simulate the scenario where a series of continuous scanning operations are needed to cover a large warehouse. We consider a simple floor plan with a straight aisle and stocking shelves beside it. A person carrying an RFID reader will move along the aisle and stop at certain locations to turn on the reader and scan the tags. Fig. 5 illustrates the setting in our simulation. Assume the reader will stop at 10 locations, $\{L1, L2, ..., L10\}$. The reading range of a reader is considered as a circle with radius R. The interval distance between any two consecutive scanning locations is set to $\frac{R}{2}$. For simplicity, we assume that the density of the products is a constant in the whole area, i.e., at any location, the number of the tags in the reading range is the same. With this setting, at each location, there are about 30% unknown tags in the reading range. The total scanning time in the whole process is compared in Table II. Compared to the CA scheme, the CU scheme dramatically reduces the scanning time by almost 50%.

CA	SK	AQS	CU
528s	562s	628s	265s

TABLE II: Total scanning time for a series of scanning operations

3) CU Extension: Finally, we evaluate the CU extension scheme, which only affects the selecting phase in the CU scheme. We present the performance in Fig. 6. Compared to the default CU scheme, the CU extension only incurs a small overhead in the selecting phase. Its performance is close to the CU scheme when U is small. Even in the worst case in Fig. 6, the extra time spent by the CU extension is less than 3 seconds.

B. Temporal Continuous Scanning

We assume that temporal continuous scans need not only collect the unknown tags, but also detect the missing tags. Thus, the implementation combines the CU and DM schemes. In our simulation, we first apply the CU scheme and then

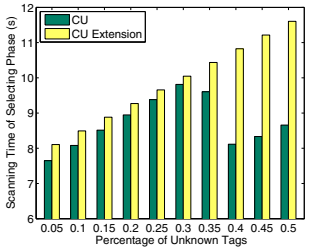


Fig. 6: Selecting Phase in the CU Extension: This figure illustrates the scanning time spent on the selecting phase in the CU extension scheme with different number of unknown tags (U). The performance of the selecting phase in the CU scheme is also displayed as a comparison. ($|\mathcal{S}| = 5000, U/|\mathcal{S}| = \{5\%, 10\%, \ldots, 50\%\}$)

the DM scheme in each scan. In the next, we present the simulation results when considering two overlapping sets. We do not consider a series of scanning in an explicit section because its performance can be easily derived from the simple case with two overlapping sets.

By default, we set $|\overline{\mathcal{S}}| = |\mathcal{S}| = 5000$ and $M = U = 0.1 \cdot |\mathcal{S}|$. The requirement for missing tags R_M is also set to $0.1 \cdot M$. In addition, the estimation of M_{max} is set to $1.2 \cdot M$.

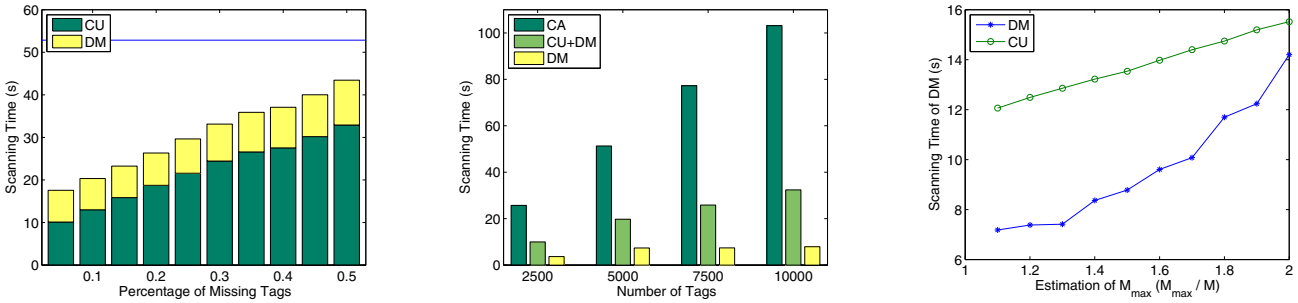
Varying M: We first test the impact of the number of missing tags M. Fig. 7(a) shows the simulation results. The horizontal axis represents the ratio of $\frac{M}{|\mathcal{S}|}$ ranging from 0.05 to 0.5 with an interval of 0.05. We separately display the scanning time consumed by the CU and DM schemes. The summary of both parts is the total scanning time in temporal continuous scanning. The performance of other three schemes is the same as that in Fig. 3(a). In Fig. 7(a), we only compare our scheme with the CA scheme whose performance appears as a flat line.

According to Fig. 7(a), we observe that our scheme for temporal continuous scanning is also very efficient. For example, in the default setting with $\frac{M}{|\mathcal{S}|} = 0.1$, our scheme consumes around 20 seconds, which is 37% of the time needed in the CA scheme. In addition, we find that the process of detecting missing tags is much shorter than collecting unknown tags and that increasing M does not incur too much time in the DM scheme. For example, the difference of DM's scanning time between $\frac{M}{|\mathcal{S}|} = 0.05$ and $\frac{M}{|\mathcal{S}|} = 0.5$ is less than 3 seconds.

Varying $|\overline{\mathcal{S}}|$ and $|\mathcal{S}|$: Similar to spatial continuous scanning, we consider 4 cases with 2500 $\sim$ 10000 RFID tags. We compare our scheme with the CA scheme in Fig. 7(b). The performance of the other two schemes (SK and AQS) is the same as in Fig. 4(a). In Fig. 7(b), we also separately illustrate the scanning time spent on the DM scheme. In Fig. 7(b), the scanning time of the DM scheme is a small overhead compared to the CU scheme and only slightly increases with the number of tags. Thus our scheme that combines CU and DM together is still much more efficient than the CA scheme, especially in a large scale system. When $|\mathcal{S}| = 10000$, our scheme consumes less than 30% of the time required by the CA scheme.

Varying M_{max}: Finally, we evaluate how the value of M_{max} affects the performance. The simulation results are presented in Fig. 7(c). The horizontal axis is the ratio of $\frac{M_{max}}{M}$ ranging from 1.1 to 2 with an interval of 0.1. As a comparison, the performance of the CU scheme is also displayed in the same

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.



(a) Varying $M(U)$: The flat line represents the performance of the CA scheme. ($|\mathcal{S}| = 5000, M = U$, and $M/|\mathcal{S}| = \{5\%, 10\%, \ldots, 50\%\}$)

(b) Varying $|\mathcal{S}|(|\overline{\mathcal{S}}|)$: The performance of the CA scheme is also presented as a comparison. ($|\overline{\mathcal{S}}| = |\mathcal{S}| = \{2500, 5000, 7500, 10000\}$)

(c) Varying estimations of M_{max}: In our settings, $M = 500$ and the ratio of M_{max}/M is set to $\{1.1, 1.2, \ldots, 2\}$.

Fig. 7: The scanning time of temporal continuous scanning with different varying parameters.

figure. Similarly, we find that more accurate estimation of M_{max} leads to a better performance. Although the absolute difference of performance is not large, e.g., the difference is less than 7 seconds between $\frac{M_{max}}{M} = 1.1$ and $\frac{M_{max}}{M} = 2$, the DM scheme is obviously more sensitive to M_{max} compared to the CU scheme.

VIII. Conclusion

This paper considers a fundamental problem of continuous scanning which often appears in RFID systems. We design efficient algorithms based on the information gathered in the previous scans and our simulation results show that the proposed schemes can dramatically reduce the scanning time.

IX. Acknowledgments

This project was supported in part by US National Science Foundation grants CNS-0721443, CNS-0831904, and CAREER Award CNS-0747108. We thank all the reviewers for their comments.

References

[1] L. Ni, Y. Liu, Y. C. Lau, and A. Patil, "Landmarc: indoor location sensing using active RFID," in *PerCom '03*.
[2] C. Wang, H. Wu, and N.-F. Tzeng, "RFID-based 3-d positioning schemes," in *INFOCOM 2007*.
[3] C.-H. Lee and C.-W. Chung, "Efficient storage scheme and query processing for supply chain management using RFID," in *SIGMOD '08*.
[4] A. Nemmaluri, M. D. Corner, and P. Shenoy, "Sherlock: automatically locating objects for humans," in *MobiSys '08*.
[5] L. Ravindranath, V. N. Padmanabhan, and P. Agrawal, "Sixthsense: RFID-based enterprise intelligence," in *MobiSys '08*.
[6] B. Metcalfe, "Steady-state analysis of a slotted and controlled ALOHA system with blocking," *SIGCOMM Comput. Commun. Rev.*, vol. 5, no. 1, pp. 24–31, 1975.
[7] F. C. Schoute, "Dynamic frame length ALOHA," *IEEE Transactions on Communications*, vol. 31, pp. 565–568, Apr. 1983.
[8] EPCglobal. Class 1 generation 2 UHF air interface protocol standard version 1.0.9. [Online]. Available: http://www.epcglobalinc.org/standards/
[9] J. Wieselthier, A. Ephremides, and L. Michaels, "An exact analysis and performance evaluation of framed ALOHA withcapture," *IEEE Transactions on Communications*, pp. 125–137, Feb. 1989.
[10] P. Hernandez, J. Sandoval, F. Puente, and F. Perez, "Mathematical model for a multiread anticollision protocol," in *IEEE Pacific Rim Conference on Communications, Computers and signal Processing*, Aug. 2001.
[11] H. Vogt, "Efficient Object Identification with Passive RFID Tags," in *International Conference on Pervasive Computing*, ser. LNCS, 2002.
[12] J. Zhai and G.-N. Wang, "An anti-collision algorithm using two-functioned estimation for RFID tags." in *ICCSA (4)*, 2005, pp. 702–711.
[13] S.-R. Lee, S.-D. Joo, and C.-W. Lee, "An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification," in *MOBIQUITOUS '05*, 2005.
[14] C. Floerkemeier and M. Wille, "Comparison of transmission schemes for framed ALOHA based RFID protocols," in *SAINT-W '06*, 2006.
[15] J.-R. Cha and J.-H. Kim, "Novel anti-collision algorithms for fast object identification in RFID system," in *ICPADS '05*, 2005.
[16] B. Zhen, M. Kobayashi, and M. Shimizu, "Framed ALOHA for multiple RFID objects identification," in *IEICE TRANSACTIONS on Communications*, 2005.
[17] M. A. Bonuccelli, F. Lonetti, and F. Martelli, "Tree slotted ALOHA: a new protocol for tag identification in RFID networks," in *WOWMOM '06*.
[18] D. Hush and C. Wood, "Analysis of tree algorithms for RFID arbitration," in *ISIT*, 1998.
[19] C. Law, K. Lee, and K.-Y. Siu, "Efficient memoryless protocol for tag identification," in *Proceedings of the 1th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, 2000.
[20] F. Zhou, C. Chen, D. Jin, C. Huang, and H. Min, "Evaluating and optimizing power consumption of anti-collision protocols for applications in RFID systems," in *ISLPED '04*, 2004.
[21] H.-S. Choi, J.-R. Cha, and J.-H. Kim, "Fast wireless anti-collision algorithm in ubiquitous id system," in *Vehicular Technology Conference*, 2004, pp. 4589–4592.
[22] A. Micic, A. Nayak, D. Simplot-Ryl, and I. Stojmenovic, "A hybrid randomized protocol for RFID tag identification," in *IEEE International Workshop on Next Generation Wireless Networks*, 2005.
[23] J. Myung and W. Lee, "An adaptive memoryless tag anti-collision protocol for RFID networks," in *IEEE ICC*, 2005.
[24] ——, "Adaptive splitting protocols for RFID tag collision arbitration," in *MobiHoc '06*, 2006.
[25] M. Kodialam and T. Nandagopal, "Fast and reliable estimation schemes in RFID systems," in *MobiCom '06*, 2006.
[26] M. Kodialam, T. Nandagopal, and W. C. Lau, "Anonymous tracking using rfid tags," in *INFOCOM 2007*.
[27] C. Qian, H. Ngan, and Y. Liu, "Cardinality estimation for large-scale RFID systems," in *PerCom '08*.
[28] M. Mitzenmacher, "Compressed bloom filters," *IEEE/ACM Trans. Netw.*, vol. 10, no. 5, pp. 604–612, 2002.
[29] S. Cohen and Y. Matias, "Spectral bloom filters," in *SIGMOD '03*.
[30] B. Chazelle, J. Kilian, R. Rubinfeld, and A. Tal, "The bloomier filter: an efficient data structure for static support lookup tables," in *SODA '04*.
[31] A. Pagh, R. Pagh, and S. S. Rao, "An optimal bloom filter replacement," in *SODA '05*.
[32] F. Bonomi, M. Mitzenmacher, R. Panigrah, S. Singh, and G. Varghese, "Beyond bloom filters: from approximate membership checks to approximate state machines," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 4, pp. 315–326, 2006.
[33] F. Hao, M. S. Kodialam, and T. V. Lakshman, "Building high accuracy bloom filters using partitioned hashing," in *SIGMETRICS '07*, 2007.
[34] NXP Semiconductors. I-code smart label RFID tags. [Online]. Available: http://www.semiconductors.philips.com/acrobat download/other/identification/SL092030.pdf